

DETAILED ACTION

EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unaccepted to applicant, an amendment may be filed as provided by 37 CFR 1,312. To ensure consideration of such an amendment, it **MUST** be submitted no longer later than the payment of the issue fee.
2. Authorization for this examiner's amendment for specification and claims 1, 12, and 23 (Cancelled claims 2-11, 13-22, and 24-33) was given in a telephone interview with Mr. Gerald H. Glanzman (Phone # (972) 385-8777 and Registration Number 25,035) for applicant on December 16, 2008.
3. The instant Examiner's amendment is directed to said entered amendment.
4. Please amend the application as follows:

IN THE SPECIFICATION

5. In the specification page 51 lines 23-31 and page 52 lines 1-4 (last paragraph of page 51 which continues in page 52) amended and needs to be **replaced** with the following paragraph:
"It is important to note that while the present invention has been described in the context of a fully functioning data processing system, those of ordinary skill in the art will appreciate that the processes of the present invention are capable of being distributed in the form of instructions in a

computer readable medium and a variety of other forms, ~~regardless of the particular type of signal-bearing media actually used to carry out the distribution.~~ Examples of computer readable media include media such as EPROM, ROM, tape, ~~paper~~-floppy disk, hard disk drive, RAM, and CD-ROMs ~~and transmission-type media, such as digital and analog communications links."~~

IN THE CLAIMS

6. Claims 1, 12, and 23 should be amended to the claim language as shown below. Claims 2-11, 13-2, and 24-33 are cancelled herein. The complete set of claims will **replace** with the claims 1-33 as filed on May 27, 2008 as follow:

1. (Currently amended) A method for operating a virtual machine within a data processing system, the method comprising the computer-implemented steps of:

a processor;

the processor for storing an object in memory that is managed by a virtual machine that is running on a device within the data processing system;

associating the object with a plurality of objects within an object group;

associating the object group with the virtual machine, wherein the virtual machine incorporates functionality for interoperating with other virtual machines in a virtual machine cluster, wherein each virtual machine in the virtual machine cluster acts as a node within the virtual machine cluster, and wherein associating the object group with the virtual machine comprises:

associating the object group with a thread managed by the virtual machine; and
storing the object group in a thread local heap associated with the thread;

managing a plurality of object groups within each virtual machine of the virtual machine cluster;

allowing the object group to be moved from the virtual machine to a different virtual machine in the virtual machine cluster;

disassociating the object group from the virtual machine, wherein disassociating the object group from the virtual machine comprises:

disassociating the object group from the thread;
terminating the thread; and

in response to terminating the thread, managing the object group without an association between the object group and the thread managed by the virtual machine;

moving the object group to the different virtual machine in the virtual machine cluster;
and

in response to moving the object group to the different virtual machine, associating the object group with the different virtual machine, wherein associating the object group with the different virtual machine comprises:

creating a first object reference for the object while the object is located on the virtual machine;

moving the object group with which the object is associated to the different virtual machine; and

creating a second object reference for the object while the object is located on the different virtual machine, wherein the first object reference and the second object reference are identical.

2-11. (Canceled)

12. (Currently amended) A computer program product on a computer readable medium for use in a data processing system for operating a virtual machine, the computer program product comprising:

a processor;

means for storing an object in memory that is managed by a virtual machine that is running on a device within the data processing system;

means for associating the object with a plurality of objects within an object group;

means for associating the object group with the virtual machine, wherein the virtual machine incorporates functionality for interoperating with other virtual machines in a virtual

machine cluster, wherein each virtual machine in the virtual machine cluster acts as a node within the virtual machine cluster, and wherein the means for associating the object group with the virtual machine comprises:

- means for associating the object group with a thread managed by the virtual machine; and

- means for storing the object group in a thread local heap associated with the thread;

- means for managing a plurality of object groups within each virtual machine of the virtual machine cluster;

- means for allowing the object group to be moved from the virtual machine to a different virtual machine in the virtual machine cluster;

- means for disassociating the object group from the virtual machine, wherein the means for disassociating the object group from the virtual machine comprises:

 - means for disassociating the object group from the thread;

 - means for terminating the thread; and

- in response to terminating the thread, managing the object group without an association between the object group and the thread managed by the virtual machine;

- means for moving the object group to the different virtual machine in the virtual machine cluster; and

- in response to moving the object group to the different virtual machine, means for associating the object group with the different virtual machine, wherein the means for associating the object group with the different virtual machine comprises:

 - means for creating a first object reference for the object while the object is located on the virtual machine;

 - means for moving the object group with which the object is associated to the different virtual machine; and

 - means for creating a second object reference for the object while the object is located on the different virtual machine, wherein the first object reference and the second object reference are identical.

23. (Currently amended) An apparatus in a data processing system for operating a virtual machine, the apparatus comprising:

a processor;

means for storing an object in memory that is managed by a virtual machine that is running on a device within the data processing system;

means for associating the object with a plurality of objects within an object group;

means for associating the object group with the virtual machine, wherein the virtual machine incorporates functionality for interoperating with other virtual machines in a virtual machine cluster, wherein each virtual machine in the virtual machine cluster acts as a node within the virtual machine cluster, and wherein the means for associating the object group with the virtual machine comprises:

means for associating the object group with a thread managed by the virtual machine; and

means for storing the object group in a thread local heap associated with the thread;

means for managing a plurality of object groups within each virtual machine of the virtual machine cluster;

means for allowing the object group to be moved from the virtual machine to a different virtual machine in the virtual machine cluster;

means for disassociating the object group from the virtual machine, wherein the means for disassociating the object group from the virtual machine comprises:

means for disassociating the object group from the thread;

means for terminating the thread; and

in response to terminating the thread, means for managing the object group without an association between the object group and the thread managed by the virtual machine;

means for moving the object group to the different virtual machine in the virtual machine cluster; and

in response to moving the object group to the different virtual machine, means for associating the object group with the different virtual machine, wherein the means for associating the object group with the different virtual machine comprises:

means for creating a first object reference for the object while the object is located on the virtual machine;

means for moving the object group with which the object is associated to the different virtual machine; and

means for creating a second object reference for the object while the object is located on the different virtual machine, wherein the first object reference and the second object reference are identical.

24-33. (Canceled)

ALLOWANCE

7. Claims 1, 12, and 23 are allowed over the prior art made of record.

REASON FOR ALLOWANCE

8. The followings are an examiner's statement of reasons for allowance:

The prior art of record fails to teach or suggest the limitations
“... disassociating the object group from the thread; terminating the thread; and in response to terminating the thread, managing the object group without an association between the object group and the thread managed by the virtual machine; moving the object group to the different virtual machine in the virtual machine cluster; creating a first object reference for the object while the object is located on the virtual machine; moving the object group with which the object is associated to the different virtual machine; and creating a second object reference for the object while the object is located on the different virtual machine, wherein the first object reference and the second object reference are identical...”

in combination with the remaining elements as cited in claims 1, 12, and 23.

9. The closest prior art, Rietschote et al. (US 7,203,944) disclose a cluster of computer systems where the computer systems is configured to execute one or more virtual machines which each computer systems comprises hardware and plurality of instruction when executed on the hardware, migrates at least a first virtual machine executing on a first computer system of second computer system of the plurality of computer systems (*See Rietschote, Col. 2, lines 11-24*), the computer system may have multiple CPUs and similarly the virtual machine may multiple CPUs too (*See Rietschote, Col. 3, lines 16-19*) also, generally, the virtual machine kernel may comprise software which may schedule virtual machines for execution on the underlying hardware, using any scheduling scheme. For example, a time division multiplexed scheme may be used to assign time slots to each virtual machine (*See Rietschote, Col. 4, lines 20-24*). However, after careful consideration of the amendment (pages 2-6) filed May 27, 2008, how the claim amendments overcome the closest prior art made of record Rietschote et al. in view of Schmalstieg et al. (US 6,842,175 B1), Sokolov et al. (US 2003/0028865 A1), and Sokolov et al. (US 6,959,430 B2) does not teach or suggest the limitations cited above as being free of any prior art when read in the claims as whole.

10. This allowable feature is indicated in independent claims 1 (Method claim), 12 (Computer program product claim), and claim 23 (Apparatus claim), provides “disassociating the object group from the thread; terminating the thread; and in response to terminating the thread, managing the object group without an association between the object group and the thread

managed by the virtual machine; moving the object group to the different virtual machine in the virtual machine cluster; creating a first object reference for the object while the object is located on the virtual machine; moving the object group with which the object is associated to the different virtual machine; and creating a second object reference for the object while the object is located on the different virtual machine, wherein the first object reference and the second object reference are identical”, as recited in independent claims1 (Method claim), 12 (Computer program product claim), and claim 23 (Apparatus claim). The closest prior art made of record Rietschote et al. in view of Schmalstieg et al. and Sokolov et al. do not disclose, teach, or suggest (in combination with all other features in the claim), the claimed limitations of method of claim 1, computer program product of claim 12, and apparatus of claim 23 as a whole. Consequently, independent claims 1, 12, and 23 as amended are allowable over prior art.

CONCLUSION

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Fariborz Khoshnoodi whose telephone number is 571-270-1005. The examiner can normally be reached on M-TH every other F 8:00-4:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Charles Ronces can be reached on 571-272-4085. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Fariborz Khoshnoodi
Examiner
Art Unit 2164

/FK/

/Charles Rones/

Supervisory Patent Examiner, Art Unit 2164